

Visión por Computadora 2024

Lista 1

18.enero.2024

1. **(No entregar)**. No hemos hablado en clase cómo leer o modificar imágenes usando OpenCV o la librería PIL.

Investigar cómo se hace la lectura de imágenes usando OpenCV y PIL en Python. Explicar cómo se almacenan estas imágenes en memoria, y explicar cómo convertir estos formatos a un array de numpy, o cómo revertir de un arreglo de numpy de vuelta a los formatos de OpenCV y PIL.

Investigar también cómo se guardan imágenes a disco usando matplotlib, OpenCV y PIL.

2. Tomar dos fotos de una misma escena pero con diferentes iluminaciones. Leer las imágenes correspondientes en Python y hacer lo siguiente:

- 1.) Construir en ambos casos los histogramas normalizados R,G,B (puede usar los 256 bins, u otra cantidad, por ejemplo 16 bins por cada canal de color).
- 2.) Proponer una distancia que permita comparar dos histogramas.
- 3.) Comparar las distancias obtenidas entre las dos fotos tomadas, y otras fotos de su elección, no relacionadas con sus fotos. Evaluar por ejemplo la distribución de las distancias obtenidas.
- 4.) Ahora, convertir cada imagen RGB a HSV.
- 5.) Construir esta vez histogramas en H,S (en 2D) y repetir el paso 3.

3. Implementar en Python una función que haga la ecualización de histograma para una imagen en escala de grises (8 bits).

Mostrar a través de ejemplos la imagen original y la imagen ecualizada. Mostrar también los histogramas normalizados y su distribución acumulada de ambas imágenes.

Comparar los resultados de su función contra alguna función ya predefinida en Python para calcular ecualizaciones. Por ejemplo, en scikit-image, tenemos la función `skimage.exposure.equalize_hist`; o en OpenCV, tenemos la función `cv2.equalizeHist`.

¿Son iguales sus resultados con los de la función? Explique por qué son iguales o explique por qué hay diferencias.

4. Implementar 3 algoritmos para segmentación binaria de imágenes. De estos, uno debe ser un algoritmo local.

Aplicar cada uno de estos algoritmos a 3 imágenes en escala de grises diferentes y comparar los resultados obtenidos de cada segmentador.

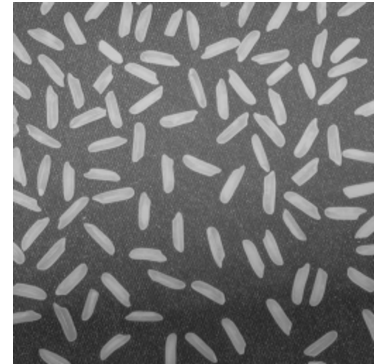
Concluya cuál de sus algoritmos obtiene mejores resultados.

5. Implementar un algoritmo de segmentación para imágenes RGB, utilizando de fondo un algoritmo de k -medias. Se debe permitir al usuario elegir el parámetro $k \geq 2$ del número de segmentos que desea obtener.

Mostrar con varios ejemplos (buenos y malos) los alcances y limitaciones de este algoritmo. Para cada caso, mostrar

- a) la imagen original,
- b) el mapa de segmentos o clases,
- c) la imagen cuantizada (promedio de color en cada segmento).

6. Aplicar transformaciones apropiadas para mejorar la visualización de las siguientes imágenes: saigon.jpeg, monkey.jpeg y rice.jpg.



Para la imagen rice.jpg, aplicar un algoritmo de segmentación binaria que identifique correctamente **todos** los granos de arroz.